TAXIS AS PERVASIVE RESILIENCE SENSORS

BY

BRIAN PATRICK DONOVAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Assistant Professor Daniel B. Work

ABSTRACT

This thesis proposes a method to quantitatively measure the resilience of transportation systems using GPS data from taxis. The granularity of the GPS data necessary for this analysis is relatively coarse; it only requires coordinates for the beginning and end of trips, the metered distance, and the total travel time. The method works by computing the historical distribution of pace (normalized travel times) between various regions of a city and measuring the pace deviations during an unusual event. This method is applied to a dataset of nearly 700 million taxi trips in New York City, which is used to analyze the transportation infrastructure resilience to Hurricane Sandy. The analysis indicates that Hurricane Sandy impacted traffic conditions for more than five days, and caused a peak delay of two minutes per mile. Practically, it identifies that the evacuation caused only minor disruptions, but significant delays were encountered during the post-disaster reentry process. Since the implementation of this method is very efficient, it could potentially be used as an online monitoring tool, representing a first step toward quantifying city scale resilience with coarse GPS data.

To my parents, for their love and support.

ACKNOWLEDGMENTS

This thesis would not have been possible without the help of many individuals. I would like to thank my advisor and mentor Daniel Work for all of his advice and support throughout all of my research. I would also like to thank the University of Illinois and Civil and Environmental Engineering departments for providing me with the education and financial support necessary to complete this research. I am specifically grateful to all of the professors in the SRIS department who encouraged me to take a risk and try a new field of research. Finally, I would like to thank all of my friends and family who have given me love, support, and encouragement to always work hard become better.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

In recent years, resilience of city infrastructure has gained a great deal of attention [3]. When disasters and other extreme events occur, infrastructure may fail, incurring large human, economic, and environmental costs. This is especially relevant for transportation infrastructure, since it is crucial for city evacuations and emergency services in post–disaster environments. Methods are needed to quantitatively monitor the transportation infrastructure in terms of its ability to withstand and recover from such events. Measuring the performance of city-scale infrastructure with traditional traffic sensors is cost–prohibitive due to relatively high installation costs, but many cities already have taxi fleets equipped with GPS sensors. Though this analysis could be performed with any GPS data, taxi data is publicly available in some cases. The New York City dataset used in this analysis gives interesting insights about the performance of infrastructure during Hurricane Sandy and other major events.

The goal of this thesis is to develop and implement a method for measuring resilience of city-scale transportation networks using only taxi datasets. The technique is designed with the following characteristics:

1. **The method can be applied at the city-scale, or larger.** Extreme events such as hurricanes have the ability to affect an entire city. For

this reason, it is important to examine impacts at a high-level city view, rather than the level of individual vehicles or streets.

2. **The method measures network performance quantitatively, in terms of recovery time and peak pace deviations.** Recovery time and peak performance degradation are fairly standard quantities of interest in the resilience literature [4, 5]. While travel times are a natural performance measure for transportation networks, we instead use *pace*, or travel time per mile. This normalization accommodates the varied length of taxi trips within a city.

3. **The method accommodates the inherent variability in traffic conditions and data.** The available data is full of noise and depends on many unmodeled human factors. As a result, the method evaluates events that cause statistically significant disruptions, in order to separate the signal from the noise.

4. **The method is computationally tractable.** Since taxi trips occur very frequently in large cities, the amount of data available for analysis is large. In order to be tractible, the computation should be $O(N)$, where N is the number of taxi trips, and ideally require only one pass through the raw data. Of practical significance, these single-pass algorithms could also be used to process the data in a realtime stream.

1.2 Related Work

In recent years, the study of resilience has gained popularity in the systems engineering community. Haimes [6, 5, 7] gives a framework for assessing resilience, which focuses on modeling a system and the possible outcomes of

various events. He asserts that a resilient system should suffer only slight degredation during an event, then rapidly recover. Reed et al. [8] note that the quality of service abruptly drops during an event, then exponentially decays back to typical values. They suggest that an appropriate resilience measure is the integral of this exponential curve. Authors in the related field of risk analysis emphasize the importance of unknown factors while assessing resilience [4, 9].

Though there is no precise consensus on the definition of resilience, *peak disruption* and *recovery time* are consistently discussed quantities. In other words, peak disruption measures how far the quantity of interest deviates from typical values, and recovery time measures how long it takes to return to typical values. Most of these works also emphasize that resilience must be measured with respect to a given event and quantity of interest. For example, one case study used the number of functioning nodes in a power grid as the quantity of interest, assessing resilience against hurricanes and minor events [10]. This thesis will follow this standard in the sense that it will use GPS data to measure the resilience of a transportation network with respect to specific events. No claims are made about the overall resilience of the network.

Several authors have proposed quantities of interest for transportation systems. Omer et al. [11] proposed a method which measures the resilience of a road-based transportation network in terms of travel times between cities. Chang et al. [12] evaluated a post-earthquake transportation network in terms of accessibility and coverage. This is partly based on an accessiblilty metric devised by Allen et al. [13], which considers travel times between various regions of a city. Thus, travel time is a standard quantity on which to measure resilience. This thesis will use the related quantity of pace, or

travel time per mile.

A distinct set of studies use large amounts of data to extract useful information about urban systems. The work most closely related to resilience is a study by He and Liu [14], which uses loop detector data to measure the effect of the I-35W bridge collapse in Minneapolis in 2007. Geroliminis et al. [15] use loop detector data, combined with 500 GPS vehicles to extract macroscopic traffic properties from an urban-scale transportation network. Other works use GPS traces of mobile devices to analyze movement patterns of crowds during typical days and atypical events [16, 17]. Castro et al. [18] present a method for inferring current and future traffic states from taxi GPS data. Zheng et al. [19] propose a method that tracks taxi trips between various regions of a city and identifies flawed urban planning. Another study measures temporal patterns in the density of taxi pickups and dropoffs to identify the social function of various city regions [20]. They point out that unusual output can be used to detect events like holidays. Chen [21] specifically focuses on identifying anomalous taxi trajectories, in order to detect fraud or special events. Ferreira et al. [22] created a graphical querying tool which can be used to count taxi trips between arbitrary geometrical regions as a function of time. They noted the drop in the frequency of taxi trips during Hurricane Sandy and Hurricane Irene, pointing out that the Irene-related drop was more significant, but the Sandy-related drop was longer lasting. By examining pace, we confirm that Hurricane Sandy had a longer recovery time, but find the contrasting result that Hurricane Sandy also has a more significant peak disruption.

1.3 Outline and Contributions

The contributions of this work are as follows. In Chapter 2, a method is proposed to use taxis as pervasive city-scale resilience sensors. This method detects unusual events and measures them in terms of peak disruption and recovery time. It introduces paces between regions of the city as the key performance measure, and it uses the historical pace distribution to detect and measure extreme events. In Chapter 3, the method is applied to a four-year dataset from New York City to identify and compare properties of events such as Hurricane Sandy. Of practical significance, the analysis identifies the relative efficiency of the pre-Sandy evacuation, contrasted with the gridlock of post-Sandy reentry. In Chapter 4, a possible extension is discussed, which automatically chooses regions in a way that examines critical infrastructure. Conclusions and future work is summarized in Chapter 5. As a technical contribution, all code [23] and data [24] used in this analysis are made publicly available.

CHAPTER 2

METHODOLOGY

2.1 Overview

The proposed technique to measure city-scale resilience of the transportation network in response to various events by examining taxi trip data is done in three steps. In section 2.2, individual taxi trips are aggregated by origin-destination pairs in order to measure typical paces between various regions of the city. This aggregation technique makes it possible to extract city-scale features at various points in time, since it is difficult to measure resilience from individual trips. Section 2.3 imposes a one-week periodic pattern on the paces, defining the mean and variance of paces for each hour of the week. Finally, Section 2.4 uses these distributions to quantify how typical or atypical the pace is at a particular point in time. Atypical paces (e.g., the 5% most unlikely points) are flagged as events, and they are examined in more detail.

2.2 Extraction of Time-Series Features from Aggregated Trips

In the first stage of analysis, trips are grouped by their geographic locations and times of occurrence. More specifically, the city is divided into a small number, k, of large regions. This allows each taxi trip to be labeled as one of

k^2 unique origin-destination pairs. Time is discretized into hours, so a large sample of trips can be gathered at any point in time. The start zone, end zone, and departure time are used to partition all of trips into subsets. The variable $T_{i,j,t}$ denotes the set of all trips from zone i to zone j at time t:

$$T_{i,j,t} = \{r | o(r) \in z(i), d(r) \in z(j), \lfloor s(r) \rfloor = t\}, \tag{2.1}$$

where $o(r)$ is the origin of trip r, $d(r)$ is the destination of trip r, $z(i)$ is the geographic region of zone i, and $\lfloor s(r) \rfloor$ is the start time of trip r, rounded down to the hour. It is assumed that i and j are both in $\{0, 1, \cdots, k-1\}$. Once these subsets of trips are defined, macroscopic traffic features can be extracted from them. Of particular interest is the expected travel time between two regions. However, travel times of individual vehicles between two regions are not uniform, due to the varying lengths of trips that connect the same regions. Much of this variation can be accounted for by normalizing against distance. In this way, the *average pace* is computed for each trip subset $T_{i,j,t}$. Trips are weighted by their distance, since longer trips give more information about the state of traffic. In this way, the distance-weighted average pace, $P(i, j, t)$, of taxis from zone i to zone j at time t is computed:

$$P(i,j,t) = \frac{\sum\limits_{r \in T_{i,j,t}} l(r)p(r)}{\sum\limits_{r \in T_{i,j,t}} l(r)} = \frac{\sum\limits_{r \in T_{i,j,t}} l(r)\frac{u(r)}{l(r)}}{\sum\limits_{r \in T_{i,j,t}} l(r)} = \frac{\sum\limits_{r \in T_{i,j,t}} u(r)}{\sum\limits_{r \in T_{i,j,t}} l(r)}, \tag{2.2}$$

where $u(r)$ is the travel time of trip r, $l(r)$ is the metered length of trip r, and $p(r) = \frac{u(r)}{l(r)}$ is the pace of trip r. For a fixed value of t, all k^2 distance-weighted average paces collectively form the *mean pace vector*, $\mathbf{a}(t)$. This vector is a function of time, and contains the k^2 pace values at a particular

point in time. Specifically, the nth element of $\mathbf{a}(t)$ is given by

$$\mathbf{a}(t)_n = P\left(\left\lfloor \frac{n}{k} \right\rfloor, n \bmod k, t\right), \tag{2.3}$$

where $n \in \{0, 1, 2, \cdots, k^2 - 1\}$.

It is desirable to use pace as the performance metric instead of the more traditional measure of vehicle counts, since the goal is to measure traffic conditions during extreme events. If the flow of vehicles between two regions drops significantly, it is difficult to determine whether this is due to increased congestion or decreased demand. However, an increase in pace indicates congestion, while a decrease in pace indicates decreased demand. Although the pace of taxis might be a biased estimate of the pace of all vehicles, logic dictates that if taxi drivers are stuck in traffic jams, so are the other vehicles around them.

2.3 Identification of City-Scale Typical Behavior

The mean pace vector, $\mathbf{a}(t)$, has a strongly periodic weekly pattern. During rush hour, the pace is high, especially in dense downtown regions, and at night the pace is low. On weekends, the rush hour is less extreme. However, the mean pace vector has some variance around this periodic pattern, so it is viewed as a distribution conditioned on time. For example, the mean pace vector for all Tuesdays at 3pm will be slightly different, and significantly different during an unusual event. To facilitate this grouping, the *reference set* Q_t is defined for all times t. This set contains all of the mean pace vectors which occur at the same point in the periodic pattern as $\mathbf{a}(t)$, except for $\mathbf{a}(t)$ itself. Intuitively, when deciding how typical the traffic data is at time t,

that data should not be used as part of the definition of typical. Since there are 168 hours in a week, the reference set can be defined as

$$Q_t = \{\mathbf{a}(h) | h \equiv t \bmod 168, h \neq t\}. \tag{2.4}$$

The reference set Q_t makes it possible to compute the expected value of the mean pace vector $\mu(t)$ as well as the covariance matrix $\mathbf{\Sigma}(t)$. This covariance matrix is important because it quantifies the noisy day-to-day fluctuations in the mean pace vector, outside of the event at hand, and how the dimensions correlate. The time-dependent sample mean and covariance matrices can be defined as:

$$\begin{aligned} \mu(t) &= \tfrac{1}{|Q_t|} \sum_{\mathbf{a} \in Q_t} \mathbf{a} \\ \mathbf{\Sigma}(t) &= \tfrac{|Q_t|}{|Q_t|-1} \left(\sum_{\mathbf{a} \in Q_t} \tfrac{\mathbf{a}\mathbf{a}^\top}{|Q_t|} - \mu(t)\mu(t)^\top \right). \end{aligned} \tag{2.5}$$

If an independence assumption is desired, the diagonal components of these matrices can be extracted. However, it is likely that many of the k^2 dimensions of $\mathbf{a}(t)$ are highly correlated, so the full covariance matrix is used for the remainder of the analysis. For example, trips that start or end in the same region often have highly correlated paces. Together, $\mu(t)$ and $\mathbf{\Sigma}(t)$ make it possible to identify unusual mean pace vectors.

2.4 Detection of Deviations from Typical Behavior

Intuitively, $\mu(t)$ captures the expected traffic conditions at a particular point in time. If the observed traffic conditions are significantly far from this expectation, then those conditions are classified as an extreme event. The covariance matrix $\mathbf{\Sigma}(t)$ is also considered; if there is typically very little deviation

from $\mu(t)$, then a large deviation is even more extreme. In one dimensional cases, this is typically addressed by standardizing the data via a z-score. In higher dimensions, the generalized z-score is called the Mahalanobis distance [25]. For this analysis, the Mahalanobis distance for an observed mean pace vector is viewed as a function of the time that the observation occurred:

$$M(t) = \sqrt{(\mathbf{a}(t) - \mu(t))^\top \mathbf{\Sigma}(t)^{-1}(\mathbf{a}(t) - \mu(t))}. \tag{2.6}$$

This time-dependent Mahalanobis distance serves as an outlier score for observations at various points in time. Note that it normalizes the deviations in each dimension by the corresponding variances, and also considers correlations between dimensions. The Mahalanobis distance is a natural way of measuring outliers in multivariate normal data, and it has shown to be useful even when the data is not normal [26]. In fact, the multivariate generalization of Chebyshev's inequality gives an upper bound on the probability of observing a Mahalanobis distance greater than some fixed value [27]. In other words, it is unlikely to observe a datapoint with a high Mahalanobis distance, regardless of the distribution. So, when $M(t)$ rises above a given threshold, an unusual event is detected. The event is declared complete when $M(t)$ returns to a value lower than the threshold. In this work, the choice of the threshold is the 95% quantile of $M(t)$, but this value can easily be lowered to detect smaller events or raised to detect only the most severe events. The function $M(t)$ is a fairly noisy, which means that it can occasionally *thrash* over the threshold. In other words, $M(t)$ may rise above the threshold, then immediately drop back below it, effectively breaking the event into two pieces. To prevent this, consecutive events separated by fewer than six hours are merged. Figure 2.1 illustrates this process.

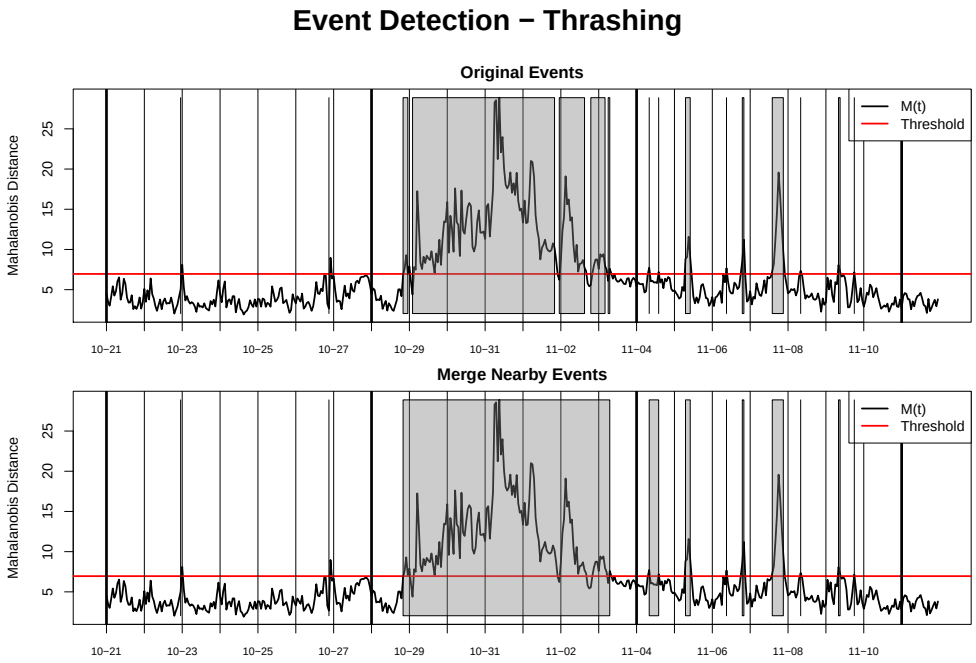

Figure 2.1: Demonstration of event detection. Events are detected when $M(t)$ goes above the threshold, but thrashing often occurs. The top graph shows that this thrashing causes events to be divided into several pieces. For this reason, events with fewer than six hours between them are merged, as shown in the bottom graph.

Once the recovery time of an event is computed, other properties can be computed. For example, it is possible to compute the maximum pace deviation, or the slowest type of trip during the event. Thus, each event can be described with a set of meaningful statistics. Comparisons between various events make it possible to describe which types of events the city can easily endure, and where there is room for improvement. For longer-lasting events like Hurricane Sandy, it is possible to examine different stages of the event in greater detail.

CHAPTER 3

APPLICATION TO HURRICANE SANDY WITH NEW YORK CITY TAXI DATA

In this section, the previously described methodology is applied to a dataset of New York City taxi trips. This dataset, which was obtained through a *Freedom of Information Law* (FOIL) request, covers four years of operation and details nearly 700 million trips. Many events are detected within this four year span and compared quantitatively. Special attention is given to Hurricane Sandy and some interesting properties are discovered.

3.1 The Dataset

The data used in this analysis takes the form of a large table where each row represents a single taxi trip. Table 3.1 gives a small sample of this data. Note that this data format is the minimum amount of information required to perform the analysis. Other datasets may contain, for example, periodic GPS updates, but this is at least as much information as the New York City data. As there are several entries per second for four years, the raw data takes up about 116GB in text CSV format. We have made this large dataset publicly available [24].

Note that this data only records trips where the taxi is occupied by a passenger. Non-occupied trips are not recorded. The dataset also contains a large number of errors. For example, there are several trips where the reported meter distances are significantly shorter than the straight-line dis-

pickup datetime	dropoff datetime	duration (sec)	distance (mi)	pickup lon	pickup lat	dropoff lon	dropoff lat
2013-05-01 00:02:11	2013-05-01 00:14:28	737	2.9	-74.00	40.74	-74.01	40.71
2013-05-01 00:02:12	2013-05-01 00:12:31	618	1.8	-74.00	40.73	-73.98	40.72
2013-05-01 00:02:12	2013-05-01 00:07:39	326	1.3	-73.97	40.76	-73.96	40.77
2013-05-01 00:02:13	2013-05-01 00:04:35	141	0.6	-73.99	40.75	-74.00	40.75
2013-05-01 00:02:14	2013-05-01 00:04:09	115	0.5	-73.98	40.75	-73.99	40.74

Table 3.1: A small subset of the data used in this analysis. Each row corresponds to an occupied taxi trip.

tance, violating Euclidean geometry. Additionally, many trips report GPS coordinates of (0,0), or contain impossible distances, times, or velocities. All of these types of obvious errors are discarded and account for roughly 7.5% of all trips.

After removing errors, the dataset is then filtered to remove data outside of the scope of the analysis. For example, there are many trips which start in Midtown, travel over 50 miles, then end less than a block from their starting points. These trips are entirely possible, but unlikely to be representative of Midtown-to-Midtown trips because they likely drove many miles in other areas. This filter is implemented by thresholding the *winding factor*, or metered distance over straight-line distance. Trips which last less than 60 seconds are also unlikely to give accurate pace estimates because the initial non-driving time becomes more important. These types of trips are also removed, accounting for roughly 4% of the original data. Figure 3.1 shows histograms of all trip features considered for filtering, as well as the thresholds used for

invalid data. Additionally, the entire months of August and September 2010 were discarded due to a high number of errors.

3.2 Computational Issues

Due to the size of the dataset, an efficient software implementation of the analysis is crucial. This section discusses the algorithmic and practical aspects of the analysis, using the New York City taxi dataset as an example. In this way, concrete figures can be used for quantities like runtime or data size. More general concepts like time complexity do not depend on the dataset.

The first step described in Section 2.2 is the most computationally expensive. All of the 697,622,444 individual trips are aggregated into 35,064 mean pace vectors - one for each hour in the four-year dataset. Since the trip data is sorted chronologically, it is possible to compute these mean pace vectors in a single pass. Recall from (2.2) that the mean pace computation involves the sum of trip durations and the sum of trip distances. Thus, these two sums are initialized to zero for each of the 16 types of trips. Each time a trip is read from the file, the relevant sums are incremented. The error filtering from Section 3.1 can also be performed at this stage, so an additional pass of the dataset is not required. When the start hour of the current trip (rounded) is greater than the start hour of the previous trip, the sums are complete for the previous hour. The mean pace vector is computed by division and output, then the sums are reset to zero. Thus, the computation is one large loop over the entire dataset. A short pseudocode is given in Algorithm 1. Note that NUM_TYPES is 16, since there are four regions.

Since each trip is accessed only once, the computation is $O(N)$, where N is the total number of trips. The computation of each hour timeslice is

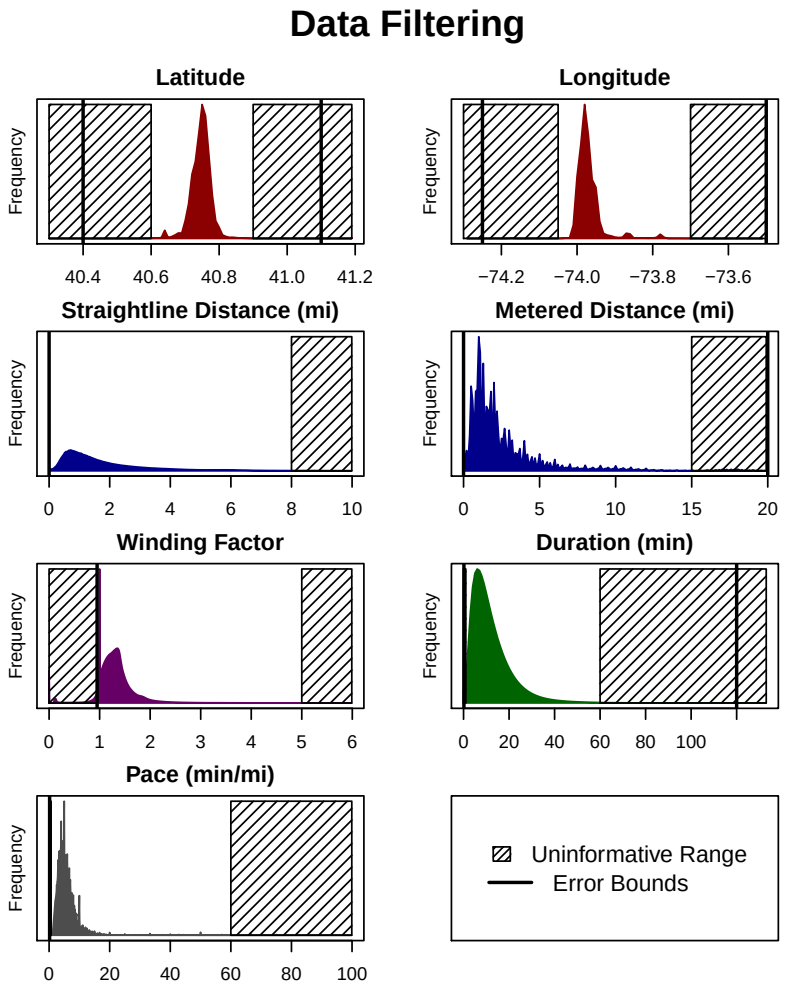

Figure 3.1: Distributions of individual features of taxi trips. Simple thresholds are used to filter trips that contain errors, or are otherwise uninformative. Note that the *straightline distance* is the Euclidean distance between start and end coordinates, while the *metered distance* is the value reported by the taximeter. The *winding factor* is the metered distance divided by the straightline distance. A winding factor less than 1 is geometrically impossible, and a large value indicates that the taxi did not proceed directly to its destination.

Algorithm 1 Online Mean Pace Vector Extraction

```
prev_hour := −1                                   ▷ Start at beginning of time
sum_duration := zeros(NUM_TYPES)                  ▷ Initialize sums to 0
sum_distance := zeros(NUM_TYPES)                  ▷ Initialize sums to 0
for all trip ∈ chronological_trips do             ▷ Loop over all trips
    while trip.hour > prev_hour do                ▷ If previous hour is complete:
        output(prev_hour, sum_duration / sum_distance)   ▷ Output mean pace vector
        sum_duration := zeros(NUM_TYPES)          ▷ Reset sums to 0
        sum_distance := zeros(NUM_TYPES)          ▷ Reset sums to 0
        prev_hour+= 1                             ▷ Advance to next hour
    end while
    if trip.isValid() then                        ▷ Data filtering
        i ← category(trip.pickup, trip.dropoff)   ▷ Determine trip type
        sum_duration[i] += trip.duration          ▷ Update distance sum
        sum_distance[i] += trip.distance          ▷ Update duration sum
    end if
end for
```

independent, making it possible to employ parallel processing if the data is partitioned ahead of time. The analysis was implemented in Python (source code available at [23]) and run on an 8-core 2.5GHz machine with 24GB of RAM. The extraction of all 35,064 mean pace vectors took about 75 minutes, using roughly 40MB of RAM for each of the eight processes. The fact that the runtime is much shorter than the real timespan of the dataset combined with the single-pass property means that this preprocessing could be performed in realtime. In other words, this system could realistically collect trips as they occur, update the relevant sums, then output the mean pace vector at the end of the hour.

The remaining computations involve mean pace vectors instead of raw trip data. They also have linear time complexity and are much faster than the preprocessing. Recall from (2.4) and (2.5) that, at a particular hour, the mean and covariance need to be computed for *all hours in the periodic pattern except that hour*. The naive implementation of this calculation has a quadratic time complexity, since each mean pace vector much be compared

against every other mean pace vector in the group. However, it is possible to compute all of these quantities in linear time. Instead of directly computing the mean of all values except $\mathbf{a}(t)$, the sum of all values including $\mathbf{a}(t)$ is computed up front. Then, in the loop, $\mathbf{a}(t)$ is subtracted from this sum. Formally, the *inclusive reference set*, Q_{t+}, is defined in a similar way to (2.4), except that it includes the mean pace vector $\mathbf{a}(t)$. In other words,

$$Q_{t+} = \{\mathbf{A}(h) | h \equiv t \bmod 168\} = Q_t \cup \{\mathbf{a}(t)\}. \tag{3.1}$$

Unlike the reference set from (2.4), the inclusive reference set is identical for values of t that occur at the same point in the periodic pattern. Thus, Q_{t+} and the sum of all vectors in Q_{t+} only need to be computed once. To compute the sum of all vectors in Q_{t+} *except* $\mathbf{a}(t)$, it is sufficient to subtract $\mathbf{a}(t)$ from this sum. Thus, the mean computation can be written as

$$\mu(t) = \frac{1}{|Q_t|} \sum_{\mathbf{a} \in Q_t} \mathbf{a} = \frac{1}{|Q_{t+}| - 1} \left(\left(\sum_{\mathbf{b} \in Q_{t+}} \mathbf{b} \right) - \mathbf{a}(t) \right). \tag{3.2}$$

A similar technique is used for the sum of outer products in the covariance computation. This method avoids redoing most of the addition in each iteration, allowing for a significant improvement on large datasets. Once $\mu(t)$ and $\mathbf{\Sigma}(t)$ are computed, $M(t)$ can be computed in constant time. Thus, the entire operation runs in linear time. On the same machine, this computation ran in less than 10 seconds, producing the timeseries of $M(t)$. Again, this operation would be feasible in a real-time system. However, it is worth noting that it may be desirable to re-generate old values of $M(t)$ in light of new information.

Once $M(t)$ is generated, the event detection described in Section 2.4 can also be performed in linear time. Events and spaces between events are

stored as a linked list, where each node contains the start time and end time. Scanning through $M(t)$ chronologically, a new node in the linked list is generated each time $M(t)$ crosses above or below the threshold. Then, to remove short spaces between events, this linked list is iterated upon. Each time a non-event node of less than the desired duration is discovered, that node and its two neighbors are replaced with one larger node. On the same machine as the previous computations, it took less than one second to perform the event detection.

3.3 Extraction of Pace Features

The map of New York City is first split into four large regions, shown in Figure 3.2. For the remainder of the analysis, the zones will be referred to in the following way: *Upper Manhattan* (U), *Midtown* (M), *Lower Manhattan* (L), and *East of the Hudson River* (E). Note that the Eastern region is connected only by bridges and tunnels and thus problems with this infrastructure will tend to increase travel times between this region and others. Specifically relevant to Hurricane Sandy is the Lower Manhattan region, since it experienced severe flooding and power outages. Choosing four large regions in this way satisfies the first goal outlined in Chapter **??** because it defines meaningful city-scale properties. Instead of looking at every street in New York under a microscope, it defines large areas with key geographic and infrastructural properties. The travel times between these regions reflect the overall performance of city-scale transportation infrastructure. It is worth noting that the methodology allows for an arbitrary choice of regions. This implementation simply chooses zones which are useful for detecting the types of events that occur in New York City.

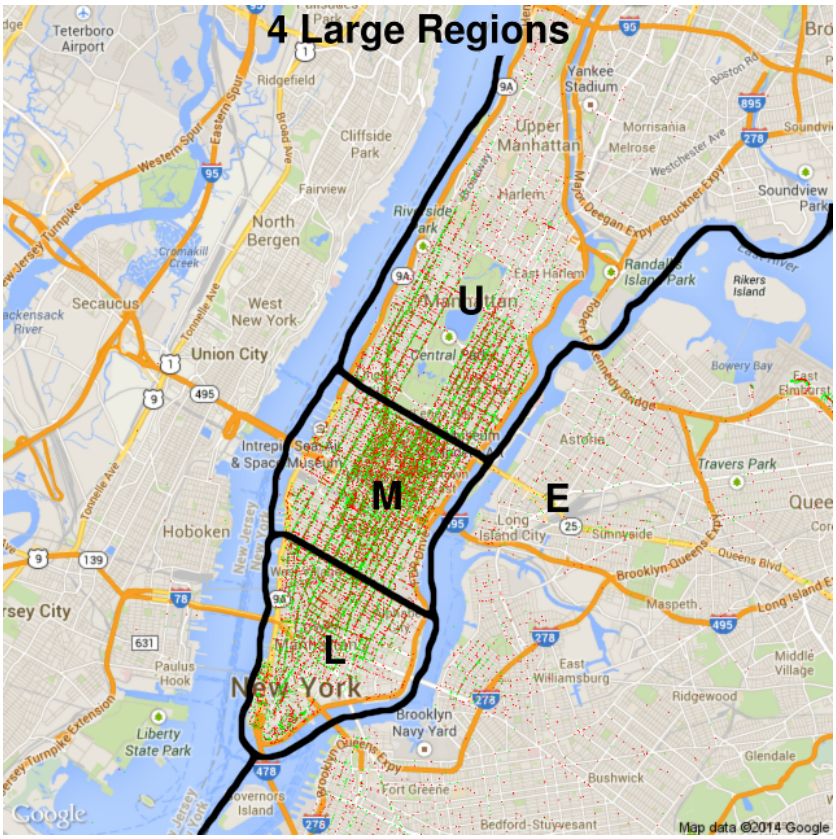

Figure 3.2: Division of New York City into four large regions denoted *U,M,E,* and *L*. A random sample of 0.01% of the taxi trips in 2012 are shown. Pickup locations are marked in green, and the corresponding dropoffs are marked in red. The majority of trips occur in Manhattan, with especially high concentration in the Midtown region.

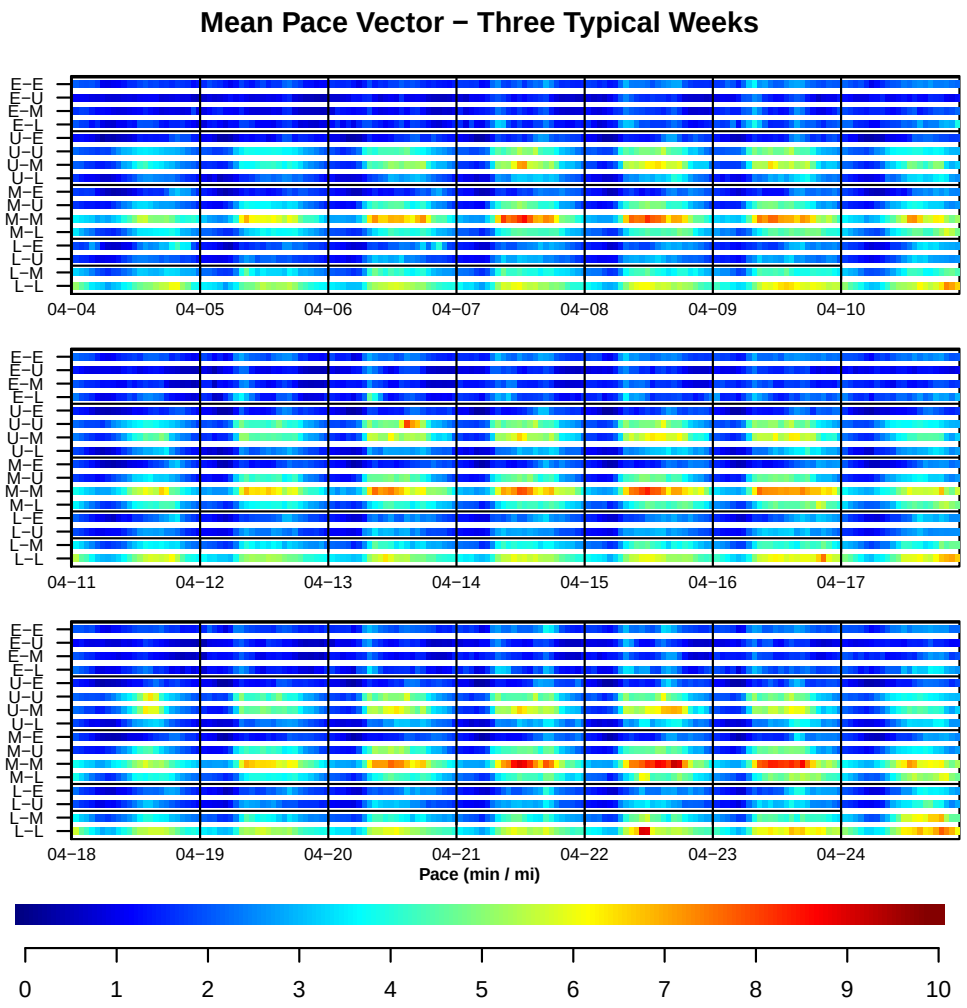

Figure 3.3: The mean pace vector, $\mathbf{a}(t)$ for three typical weeks, starting on April 4, 2010. A periodic pattern is observable, with high paces during rush hour.

Recall that a taxi can take one of 16 possible trips between these regions. Aggregating these trips by type and hour as in Section 2.2 produces the 16-dimensional mean pace vector, $\mathbf{a}(t)$, at all points in time. Figure 3.3 shows three typical weeks of mean pace vectors, revealing the expected periodic pattern.

3.4 Analysis of Events

As detailed in Section 2.3, the expected behavior is generated for all times t according to $\mu(t)$ and $\mathbf{\Sigma}(t)$. An interesting way to view the mean pace vector $\mathbf{A}(t)$ is by standardizing it, element by element, producing the *standardized pace vector*. The ith element of this vector is given by

$$\mathbf{S}(t)_i = \frac{\mathbf{A}(t)_i - \mu(t)_i}{\sqrt{\mathbf{\Sigma}(t)_{i,i}}}. \tag{3.3}$$

Intuitively, the standardized pace vector tells how many standard deviations away from the mean the pace of each category of trips is at time t. In other words, it is possible to identify the trips that are going slower or faster than expected, and how significant this difference is. Figure 3.4 shows the standardized pace vector during the week of Hurricane Sandy. This figure gives some intuition on the behavior of various regions of the city during and after the hurricane. It also includes labels indicating the occurrences of various phases of the event, obtained from a post-Hurricane Sandy study from NYU [2]. Standardizing each origin-destination pace separately allows for additional insight beyond the Mahalanobis distance.

The most notable finding is that the slowest traffic occurred on Wednesday October 31st, almost two days after the hurricane struck land. On this day, some airports, buses, and commuter rails attempted to resume normal service, but much of the infrastructure was still damaged [2]. It is even more surprising that Midtown-to-Lower Manhattan and Lower Manhattan-to-Lower Manhattan travel times are significantly *lower* than expected during this time. The pace of these trips remains almost five standard deviations below the mean until Saturday the third, despite the severe flooding and power outages in Lower Manhattan. The fact that a hurricane can actually

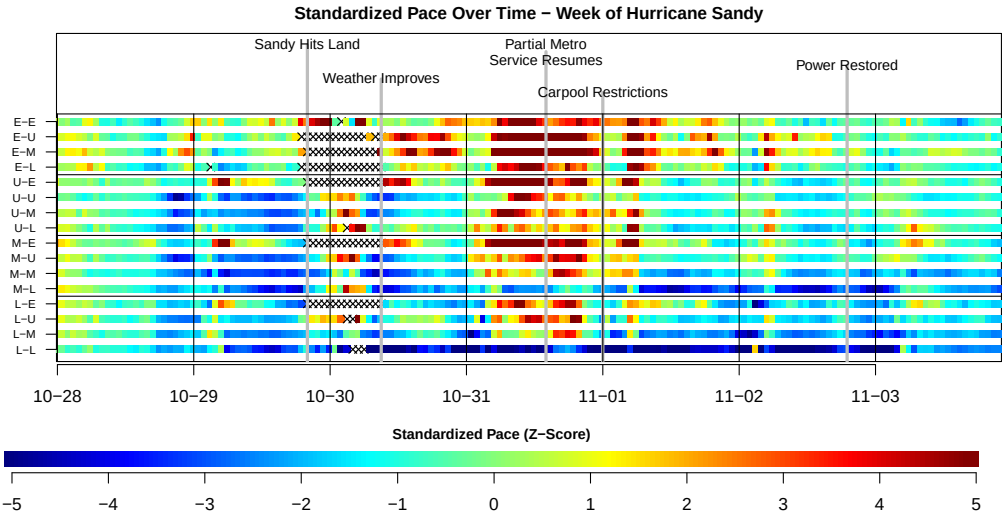

Figure 3.4: The standardized pace vector during the week of Hurricane Sandy, 2012. Labels are included to show the times of specific phases of the event [2]. An average week would have values of zero everywhere, but significant deviations are shown during the week of Hurricane Sandy. Missing data (hours where there are less than five occurrences of a given trip) are marked with black Xs.

make traffic move faster in some areas of the city indicates that the usage of the infrastructure changed. It is likely that the hurricane decreased demand on the transportation network in Lower Manhattan until the infrastructure began to recover.

This standardized pace vector gives a meaningful interpretation of unusual travel times between various regions of the city, but it fails to account for correlations between these typical travel times i.e., the off-diagonal elements of $\mathbf{\Sigma}(t)$. In contrast, the Mahalanobis distance $M(t)$ considers the full covariance matrix. As described in Section 2.4, events are detected when $M(t)$ goes above a threshold for a significant period of time. Figure 3.5 shows this process, along with the average pace of all taxis. Table 3.2 shows the top ten events, sorted by duration. At the top of the list is Hurricane Sandy, taking over five and a half days for travel times to return to normal. This is over three times the recovery time of Hurricane Irene. This agrees with the results

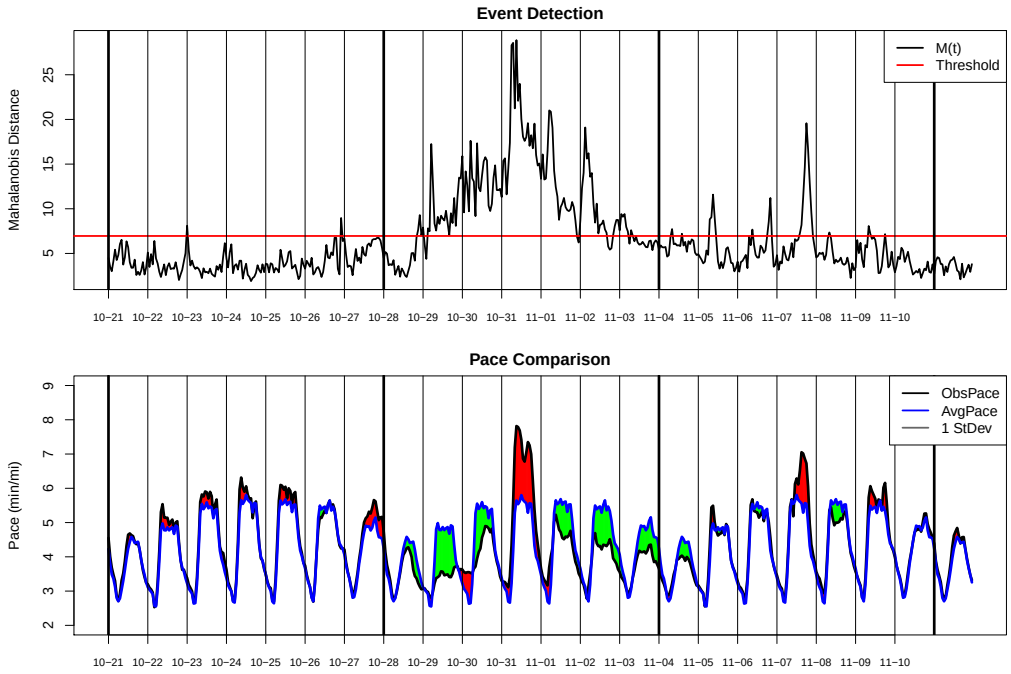

Figure 3.5: Probabilistic detection and measurement of the event Hurricane Sandy. The Mahalanobis distance, $M(t)$, is plotted in the top figure and events are detected when it goes below the threshold. For comparison, the average pace of all taxis in the city is plotted below and compared to the expected value. Green areas indicate that travel times are low, but red indicates that they are unusually high.

of [22], which showed that the total number of Manhattan taxi trips returned to normal more quickly during Hurricane Irene than Hurricane Sandy. At its worst, Sandy added over two minutes to each mile driven by taxis in the city, while Irene added less than forty seconds. This is in contrast to the results of [22], which showed that the peak drop in the number of taxi trips was greater during Hurricane Irene. The blizzard of December 2010, while shorter, added four minutes of travel time to each mile at its peak.

It is difficult to evaluate the accuracy of the results in Table 3.2, since the true severity of each event is not known. If a training set of events is available, one could raise or lower the detection threshold until the desired balance between type I and type II errors is reached.

Event	Start Time	Duration (hours)	Max (min/mi)	Min (min/mi)	Worst Trip
Sandy	2012-10-28 21:00:00	132	2.25	-1.6	E $\rightarrow$ M
Blizzard	2010-12-26 13:00:00	112	4.41	0.33	M $\rightarrow$ M
Blizzard	2011-01-31 08:00:00	49	2.04	0.34	E $\rightarrow$ E
Irene	2011-08-27 13:00:00	43	0.64	-1.66	E $\rightarrow$ E
Unknown	2013-10-12 03:00:00	33	1.09	0.08	E $\rightarrow$ L
Blizzard	2013-02-08 06:00:00	26	1.54	-0.58	E $\rightarrow$ E
Blizzard	2010-02-10 06:00:00	24	0.67	-1.01	E $\rightarrow$ E
New Years	2012-12-31 15:00:00	20	1.42	-2.66	E $\rightarrow$ M
Unknown	2011-09-09 08:00:00	19	1.66	0.35	U $\rightarrow$ U
Blizzard	2011-01-28 02:00:00	18	2.57	0.49	L $\rightarrow$ L

Table 3.2: Comparison of New York City transportation infrastructure resilience to the 10 longest events. The duration in hours, and the maximum/minimum pace deviation in minutes/mile is given for each event. Note that a positive number indicates a delay while a negative indicates a decreased pace. The final column indicates which of the 16 trips most frequently had the highest standardized pace during the event. Labels for events (the first column) are determined manually (cf. [1]).

CHAPTER 4

CHOOSING REGIONS

4.1 Motivation

The methodology explained and tested in the previous chapters measures the average pace of taxis between various regions of a city and uses these paces to identify abnormal traffic conditions. Though it is possible to apply the methodology to regions of any shape, size, and quantity, the given application uses four hand-drawn regions. For many applications, it is desirable to choose regions automatically, in a way that considers the structure of the road network. This section explains an extension to the methodology which identifies critical infrastructure, and defines regions as areas which are only connected by this infrastructure. For example, two islands which are only connected by a small number of bridges should be separated into two distinct regions. Computationally, this extension makes use of recent advances in graph-partitioning algorithms.

Intuitively, choosing regions in this way can make the analysis of extreme events more meaningful for two reasons. First, the degradation or failure of critical infrastructure is likely during extreme events. This can have disastrous effects on travel times between various regions of the city. For example, during Hurricane Sandy, many of the bridges connecting Manhattan to the mainland were closed. This has a much more significant effect than if, for example, a few residential streets were closed. Second, this choice of regions

makes it easier to pinpoint the cause of failures, since slow travel times between two adjacent regions indicate a failure in the critical infrastructure that connects them.

4.2 Methods

The identification of critical infrastructure makes use of the concept of minimum cuts from graph theory. When the nodes of a graph are partitioned into two distinct sets, the edges that connect these two sets are the *cut edges*. A *minimum cut* is a partitioning that has the fewest number of cut edges. However, for most purposes including ours, minimum cuts are not able to produce a meaningful partition, since the resulting regions will have unbalanced sizes. Indeed, an easy way to generate a small number of cut edges is to cut off only one node. Many graph partitioning algorithms in the literature attempt to combat this issue.

Spectral clustering is a popular algorithm for clustering both graphs and tabular data. The goal of this algorithm is to partition a graph into k pieces of roughly the same size while cutting a small number of edges. There are several versions of this algorithm, one of which approximately minimizes the *RatioCut* criterion [28]:

$$RatioCut = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{|A_i|}, \tag{4.1}$$

where A_i is the set of nodes in region i, $\bar{A}_i$ is the set of nodes *not* in region i, and $cut(A_i, \bar{A}_i)$ is the number of cut edges that touch a node in region i. Intuitively, this objective function is a tradeoff between minimizing the number of cut edges and keeping all of the regions large. Other criteria, such as *Ncut*, measure the region size by the number of edges.

While these methods are successful for many types of graphs, they fail to produce reasonable clusters for road networks. The problem is that road networks are extremely sparse, due to geometrical constraints. Thus, almost any cut will cut a relatively small number of edges. In practice, when clustering road networks, spectral clustering tends to focus more on producing uniformly-sized regions than cutting a small number of edges. Figure 4.1 shows the results of applying spectral clustering to the New York City graph, using four and ten clusters. The four-cluster results are poor, as the algorithm fails to divide the island of Manhattan from the mainland. Equivalently, it fails to identify the critical infrastructure by cutting the small number of bridges that connect these regions. When $k = 10$, the results are slightly better. However, examining the boundaries between regions two and nine show that algorithm failed to cut the key bridges that connect the northern part of Manhattan to the Bronx.

For road networks, a better solution is to *constrain* the sizes of the regions. The KaFFPa algorithm requires that all of the regions are no larger than a fixed size, while minimizing the size of the cut [29]. Practically, this ensures that the region sizes are approximately balanced. It is a multi-level graph partitioning algorithm, meaning that it first contracts many of the nodes into blocks, performs the partitioning on this simpler graph, then projects the results back to the original graph. The power of the algorithm comes from a refinement step, where max-flow relationships between adjacent blocks are used to reassign nodes and further decrease the cut size. Improvements to this algorithm make use of parallel processing and randomized search techniques to iteratively improve the partitioning [30].

In order to partition the graph of New York City, we made use of the KaFFPaE software by the same authors [31]. Figure 4.2 shows the partition-

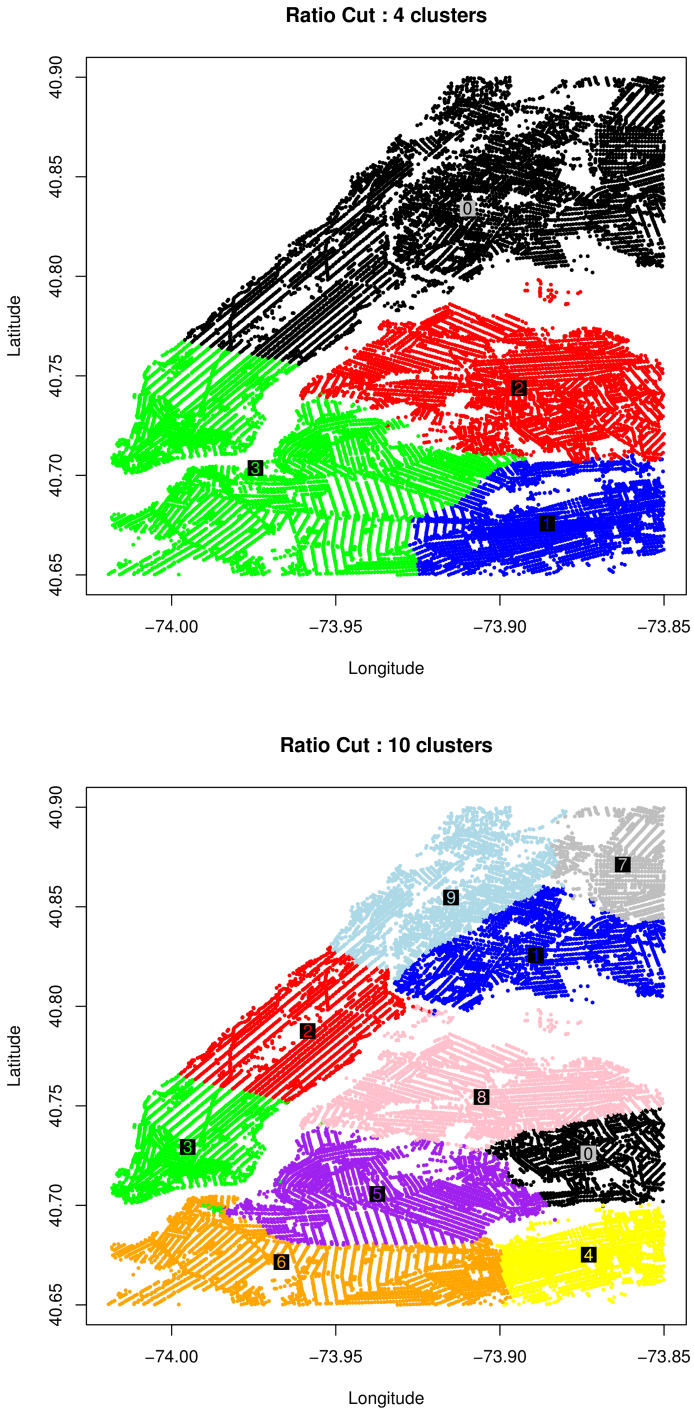

Figure 4.1: Regions produced by spectral clustering using four and ten partitions. The algorithm fails to identify critical infrastructure and regions span across bodies of water.

ing results on the same graph, using four and ten regions, and a maximum of 20% imbalance between region sizes. Clearly, the partitions are much higher quality than the ones produced by spectral clustering. In particular, it accurately separates the island of Manhattan from the rest of the graph. This means that it has correctly identified the bridges as critical infrastructure.

This chapter demonstrates how the KaFFPaE software could be used to partition city-scale road networks, in order to automate the choice of regions which are drawn by hand in Chapter 3. In particular, it defines regions so they are sparsely connected by critical infrastructure like bridges, tunnels, and highways. Thus, the outlier detection techniques described in 2 will place extra emphasis on the performance of this key infrastructure. This extension provides a much clearer procedure for reproducing the methodology in other cities.

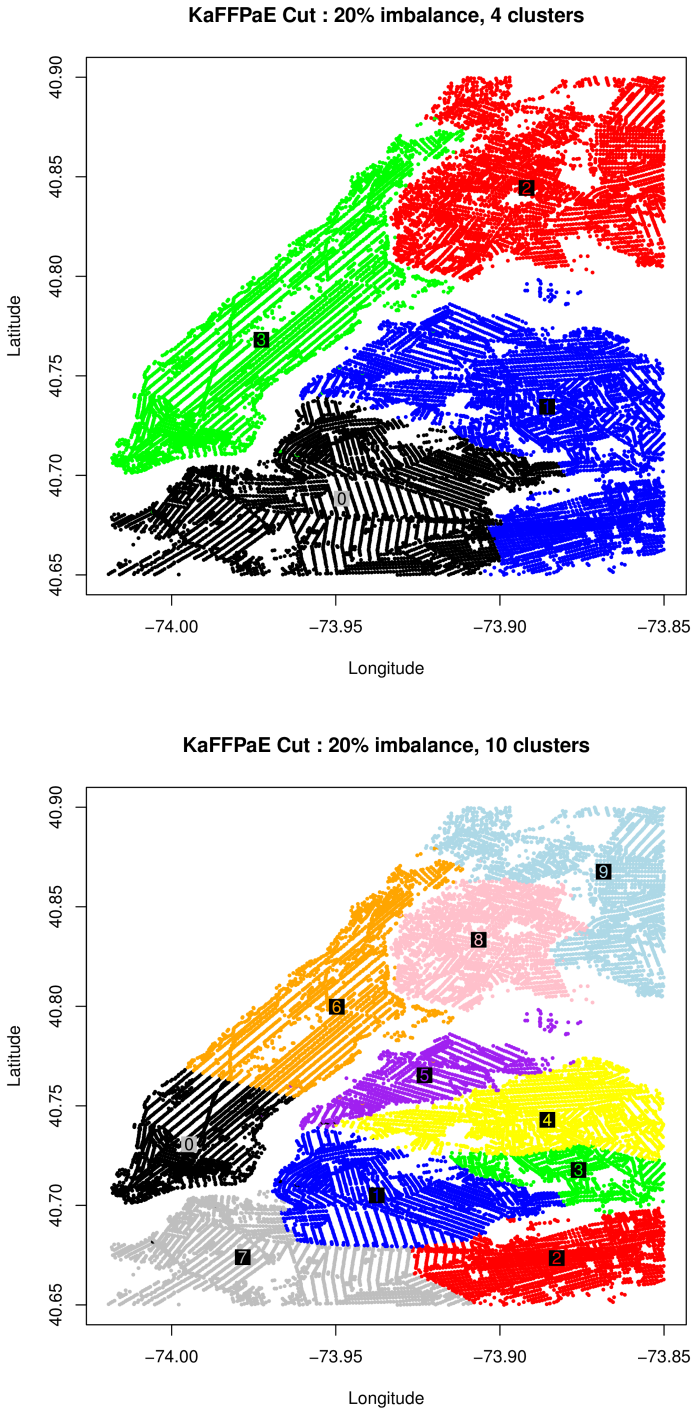

Figure 4.2: Regions produced by the KaFFPaE software package. The algorithm successfully cuts bridges, which are the critical infrastructure connecting the island of Manhattan to other regions.

CHAPTER 5

CONCLUSION

This analysis has shown that it is possible to detect and measure the effects of unusual events on transportation infrastructure using only taxi GPS data. This is a first step toward assessing and improving city-scale resilience. Of key importance, the method is extremely low cost, because it does not require the installation of any additional sensors. This method proposes computing *origin-destination paces*, or average travel time per mile between various regions of the city. The effects of various events are quantified by the sizes and durations of pace deviations from typical values. Importantly, this measurement considers the typical statistics of traffic conditions, so significant events can be distinguished from random day-to-day fluctuations.

The proposed method is applied to a dataset from New York City, and Hurricane Sandy is analyzed in detail. The analysis shows this was the longest event in the four year dataset, and one of the most severe in terms of peak pace deviation. At its worst, Hurricane Sandy caused over two minutes of delay per mile, but actually resulted in *faster* traffic for most of its duration. Most interestingly, the spike in delay occurred two days after the hurricane struck, as many residents migrated back into the city. This re-entry process was extremely slow when compared to the evacuation process before the hurricane, suggesting that more traffic management might be necessary following an event. The analysis of an extreme event like Hurricane Sandy demonstrates the ability of the proposed method to capture and describe

atypical city-scale properties of the transportation network.

5.1 Future Work

This research is ongoing, and leaves several opportunities for improvement. For example, instead of measuring paces between various origin-destination zones, one may desire to compute approximate paces on each link of the network graph. Algorithms exist which can estimate link travel times, for example [32] and [33], but they are computationally expensive. If the same outlier-detection methods are applied to link-level pace data, it is possible to examine whether such a heavy computation is necessary. If the results are unchanged, the simpler method presented in this thesis may be sufficient.

Others may note that the leave-one-out Mahalanobis distance may not be an appropriate outlier measure when many outliers are present in the dataset. Extreme outliers tend to skew the estimate of the covariance matrix, which can make it difficult to identify the *other* outliers. This phenomenon is especially true in high dimensional data, which occurs when using many regions or the link-level paces previously mentioned. Recent convex optimization techniques make it possible to discover low-rank approximations of high-dimensional data which are robust to the presence of outliers [34]. Outliers can then be detected in this lower dimensional space. The use of these techniques has the potential to strengthen the analysis, especially if link-level data is used.

REFERENCES

[1] "Lmgtfy," http://lmgtfy.com/?q=new+york+city+october+28+2012.

[2] S. Kaufman, C. Qing, N. Levenson, and M. Hanson, "Transportation during and after Hurricane Sandy," Rudin Center for Transportation, NYU Wagner Graduate School of Public Service, Tech. Rep., 2012.

[3] D. Matherly and N. Langdon, "A guide to regional transportation planning for disasters, emergencies, and significant events," National Cooperative Highway Research Program, Tech. Rep. 777, 2014.

[4] T. Aven, "On some recent definitions and analysis frameworks for risk, vulnerability, and resilience," *Risk Analysis*, vol. 31, no. 4, pp. 515–522, 2011.

[5] Y. Y. Haimes, "On the definition of resilience in systems," *Risk Analysis*, vol. 29, no. 4, pp. 498–501, 2009.

[6] Y. Y. Haimes, "On the complex definition of risk: A systems-based approach," *Risk Analysis*, vol. 29, no. 12, pp. 1647–1654, 2009.

[7] Y. Y. Haimes, "Responses to Terje Aven's paper: On some recent definitions and analysis frameworks for risk, vulnerability, and resilience," *Risk Analysis*, vol. 31, no. 5, pp. 689–692, 2011.

[8] D. A. Reed, K. C. Kapur, and R. D. Christie, "Methodology for assessing the resilience of networked infrastructure," *IEEE Systems Journal*, vol. 3, no. 2, pp. 174–180, 2009.

[9] S. Kaplan and B. J. Garrick, "On the quantitative definition of risk," *Risk Analysis*, vol. 1, no. 1, pp. 11–27, 1981.

[10] M. Ouyang, L. Dueñas-Osorio, and X. Min, "A three-stage resilience analysis framework for urban infrastructure systems," *Structural Safety*, vol. 36–37, pp. 23–31, 2012.

[11] M. Omer, A. Mostashari, and R. Nilchiani, "Assessing resilience in a regional road-based transportation network," *International Journal of Industrial and Systems Engineering*, vol. 13, no. 4, pp. 389–408, 2013.

[12] S. E. Chang and N. Nojima, "Measuring post-disaster transportation system performance: the 1995 Kobe earthquake in comparative perspective," *Transportation Research Part A: Policy and Practice*, vol. 35, no. 6, pp. 475–494, 2001.

[13] W. B. Allen, D. Liu, and S. Singer, "Accesibility measures of US metropolitan areas," *Transportation Research Part B: Methodological*, vol. 27, no. 6, pp. 439–449, 1993.

[14] X. He and H. X. Liu, "Modeling the day-to-day traffic evolution process after an unexpected network disruption," *Transportation Research Part B: Methodological*, vol. 46, no. 1, pp. 50–71, 2012.

[15] N. Geroliminis and C. F. Daganzo, "Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings," *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 759–770, 2008.

[16] F. Calabrese, F. C. Pereira, G. D. Lorenzo, L. Liu, and C. Ratti, "The geography of taste: analyzing cell-phone mobility and social events," in *Proceedings of the 8th International Conference on Pervasive Computing*. Springer, 2010, pp. 22–37.

[17] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cell phones: A case study in Rome," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 141–151, 2011.

[18] P. S. Castro, D. Zhang, and S. Li, "Urban traffic modelling and prediction using large scale taxi GPS traces," in *Proceedings of the 10th International conference on Pervasive Computing*. Springer, 2012, pp. 57–72.

[19] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proceedings of the 13th International Conference on Ubiquitous Computing*. ACM, 2011, pp. 89–98.

[20] G. Qi, X. Li, S. Li, G. Pan, Z. Wang, and D. Zhang, "Measuring social functions of city regions from large-scale taxi behaviors," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011*, 2011, pp. 384–388.

[21] C. Chen, D. Zhang, P. Samuel Castro, N. Li, L. Sun, and S. Li, "Real-time detection of anomalous taxi trajectories from gps traces," in *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, A. Puiatti and T. Gu, Eds. Springer Berlin Heidelberg, 2012, vol. 104, pp. 63–74.

[22] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual exploration of big spatio-temporal urban data: A study of New York City taxi trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, 2013.

[23] B. Donovan, "Published code," https://github.com/Lab-Work/gpsresilience.

[24] B. Donovan and D. Work, "New York City taxi data 2010–2013," http://publish.illinois.edu/dbwork/open-data/.

[25] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[26] R. Warren, R. F. Smith, and A. K. Cybenko, "Use of mahalanobis distance for detecting outliers and outlier clusters in markedly non-normal data: A vehicular traffic example," DTIC Document, Tech. Rep., 2011.

[27] J. Navarro, "Can the bounds in the multivariate Chebyshev inequality be attained?" *Statistics & Probability Letters*, vol. 91, pp. 1–5, 2014.

[28] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[29] P. Sanders and C. Schulz, "Engineering multilevel graph partitioning algorithms," in *Algorithms–ESA 2011*. Springer, 2011, pp. 469–480.

[30] P. Sanders and C. Schulz, "Distributed evolutionary graph partitioning." in *Proceedings of the Fourteenth Workshop on Algorithm Engineering and Experiments*. SIAM, 2012, pp. 16–29.

[31] C. Schulz, "Published code," https://github.com/schulzchristian/KaHIP/.

[32] T. Hunter, R. Herring, P. Abbeel, and A. Bayen, "Path and travel time inference from GPS probe vehicle data," *NIPS Analyzing Networks and Learning with Graphs*, 2009.

[33] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 37, pp. 13 290–13 294, 2014.

[34] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," in *Proceedings of the Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.